Theoretical Computer Science 450 (2012) 109–116

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Using minimal absent words to build phylogeny

Supaporn Chairungsee [a,*], Maxime Crochemore [a,b]

[a] King's College London, London WC2R 2LS, United Kingdom
[b] Université Paris-Est, France

ARTICLE INFO

Keywords:
Minimal absent words
Forbidden words
Trie of bounded depth
String similarity
Phylogeny construction

ABSTRACT

An absent word in a sequence is a segment that does not occur in the given sequence. It is a *minimal absent word* if all its proper factors occur in the given sequence.

In this paper, we review the concept of minimal absent words, which includes the notion of shortest absent word. We present an efficient method for computing the minimal absent words of bounded length using a trie of bounded depth, representing bounded length factors. This method produces the minimal absent words of given bounded length, and furthermore our technique provides a linear-time algorithm with less memory usage than previous solutions. We also present an approach, the length-weighted index, to distinguish sequences of different organisms using their minimal absent words. The results show that we can build a phylogenetic tree based on the information collected.

1. Introduction

Processing DNA sequences in an efficient way is a fundamental precondition for the study and analysis of biological molecules; see, for example, [4]. An alignment method is a method to align given strings (sequences) by inserting possible gaps and by allowing mismatches until the strings have the same length [4]. Alignment methods play an important role in molecular biology because they are used, for instance, for sequence comparisons, for subsequence searching, or for motif inference. Moreover, alignment methods are useful for the comparison of DNA sequences and for protein sequence comparisons of different species in order to compare the species [12]. The similarity/dissimilarity information provided by this approach can be used to determine the relationship of these species.

In molecular biology, the similarity/dissimilarity of the sequences between organisms is useful information for building phylogenies. A phylogeny (also called a cladogram or a dendogram in other contexts) is represented by a leaf-labelled tree [13]. This tree contains leaves that correspond to the existing taxa and internal nodes that stand for hypothetical ancestors of the taxa. It is assumed that two similar taxa will be represented as neighboring external branches and will be joined to a common parent branch. An interesting problem for sequence similarity/dissimilarity measures is selecting suitable invariants/descriptors to characterize DNA sequences of species for sequence comparisons in an efficient way rather than using their whole genomes. A standard example for DNA analysis is to select the first axon of the β-globin gene, because the gene family of β-globin has a significant biological role in oxygen transport in organisms.

An absent word in a molecular sequence is a word that does not appear in the given sequence. These words are assumed to refer to negative selection. They can be used as biomarkers for preventive and curative medical applications that are derived from personal genomic efforts. If absent words can be identified, this information will be useful for genome comparison, genetic engineering, and sequence evolution. For instance, aa, bbb, ababa, ababb, and bbabb are the minimal absent words of a string $y = \texttt{babbabab}$. The similarity/dissimilarity measure between genomic sequences provides important

* Corresponding author.
E-mail addresses: supaporn.chairungsee@kcl.ac.uk (S. Chairungsee), maxime.crochemore@kcl.ac.uk (M. Crochemore).

0304-3975/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2012.04.031

information for phylogeny construction based on distance. The gene family of β-globin is used to analyze DNA and the first axon of the β-globin gene is an example for many DNA studies instead of computing the similarity/dissimilarity of the whole genomes.

In this paper, we present two efficient approaches to construct a phylogenetic tree based on minimal absent words. First of all, we review the concept of minimal absent words, which includes the notion of shortest absent words. In order to compute the minimal absent words efficiently, we present a new approach to find the minimal absent words of given length of the input sequence using the trie of its bounded length factors. This approach consumes less memory space than previous approaches, while it can compute the bounded set of minimal absent words. Moreover, our method runs in linear time according to the sequence length. Second, we introduce a solution to discriminate genomic sequences using their minimal absent words. We define the notion of a length-weighted index to compute the similarity/dissimilarity between sequences based on the minimal absent words. Finally, we apply the technique to the first axon of β-globin. It is used to build a phylogeny of the 11 organisms aforementioned. This confirms the result obtained in [10], and proves that our approach is valid.

The first study of absent words was presented by Béal et al. [3], and their work demonstrated that the growth rate of the set of minimal absent words for a factorial formal language was changeless. In 1998, Crochemore et al. [6] presented the computation for all the forbidden words of a sequence in linear time and in linear memory space. Their work could produce the trie of minimal forbidden words from the factor automaton of a single word. In 2000, Crochemore et al. [7] presented antidictionary, a new approach for text compression based on forbidden words. Their method runs in linear time, and has even been extended to regular languages by Béal and Crochemore [2]. Hampikian and Andersen defined the term nullomer to refer to the shortest words that do not occur in a given genome and the term prime to refer to the shortest words that are absent from the entire known genetic data [8]. Their motivation was to discover the constraints on natural DNA and protein sequences. Their algorithm could track the occurrences of all absent possible words to a user-specified length limit n, using a set of 4^n counters for the 4^n possible words of length n. This produces the existing absent words up to the given length limit n. In the same year, Acquisti et al. [1] studied nullomers and the cause of absent words in the human genome. In 2008, the term unwords, the shortest absent words, was invented by Herold et al. [9]. They presented a method to compute the shortest absent words, and their approach runs in linear time. The drawback of their work is it can produce only the shortest absent words. Pinho et al. defined the term minimal absent words, which form a set smaller than the set of absent words. This term includes the shortest absent words [11]. They also designed an algorithm to produce the minimal absent words. Their algorithm uses a suffix array technique to compute the minimal absent words. The running time of the algorithm is not linear. In 2010, the algorithm to compute the shortest absent words using a probabilistic method was presented by Wu et al. [14]. Their algorithm runs in linear time, and this approach requires less memory than the previous methods.

2. Basic definition

In this section, we first recall the notions of factor, absent words, minimal absent words, and the unweighted pair group method with arithmetic mean (UPGMA). Next, we define the trie of bounded length factors and the length-weighted index.

A string x is a factor of a string y if there exist two strings u and v such that $y = uxv$ [5]. For example, string $x = \texttt{aba}$ is a factor of the string $y = \texttt{aabbababba}$.

An absent word (also called an unword or a forbidden word in other contexts) in a string is a word that does not occur in the given string. String u is an absent word in a string y if it is not a factor of y, i.e., it does not appear in y. For instance, aaa, aba, and bbb are examples of absent words for the string $y = \texttt{abbaab}$.

An absent string u is said to be minimal absent word if all its proper factors are factors of y. For example, aaa, aabb, aba, bab, and bbb are minimal absent words of string $y = \texttt{abbaab}$.

The length-weighted index provides a measure of the similarity/dissimilarity of sample sets by considering the length of each member in the symmetric difference ($X \triangle Y$) of the sample sets. For sample sets X and Y, we define this index to be

$$L - W(X, Y) = \sum_{w \in X \triangle Y} \frac{1}{|w|^2}.$$

For example, let us consider two given sets X and Y, where $X = \{\texttt{aaa, aabb, aba, bbab, bbb}\}$, and $Y = \{\texttt{aa, aba, baba, bbb}\}$. The set of $X \triangle Y$ is $\{\texttt{aa, aaa, aabb, baba, bbab}\}$. The length-weighted index of this set is $\frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{4^2} + \frac{1}{4^2}$, which is 0.548611.

The UPGMA is a technique to reconstruct the phylogenetic tree, T, for a set S of n taxa [13]. The basic principle of the UPGMA is that similar taxa should be close in the phylogenetic tree. Therefore, this technique builds the tree by clustering similar taxa iteratively, and it works by building the phylogenetic tree bottom up from its leaves.

The trie of bounded length factors of a string is the deterministic automaton that recognizes the set of fixed length factors of the string. We consider a string y of positive length n on the alphabet $A : y = y[0\,..\,n-1]$. Let k denote the length of factors and let w denote the set of bounded length factors of a string y. The set w is defined by

$$w = \{y[i\,..\,k-1+i] | k-1+i < n,\ 0 \le i < n, 0 \le k < n\}.$$

For instance, let us consider the string $y = \texttt{CAGACCGTTT}$. The bounded length factor value will be 4. The factors of y with bounded length factor 4 are CAGA, AGAC, GACC, ACCG, CCGT, CGTT, GTTT, TTT, TT and T. Fig. 1 displays the trie of bounded length factors is equal to 4 of this string.

3. Method

In this section, we first describe how to compute minimal absent words with the trie of bounded length factors in Section 3.1. Then we present the similarity/dissimilarity measure with the length-weighted index in Section 3.2.

3.1. Minimal absent words trie computation

In this subsection, we present how to compute the minimal absent words of a string in linear time with the bounded length factors trie. The code of the algorithm below uses the trie of the string y with the bounded length ℓ, $\mathcal{T}(y, \ell)$. The algorithm works as follows.

At a given step, *List* is a queue to store pairs of nodes of $\mathcal{T}(y, \ell)$ and nodes of the minimal absent words trie of the string y, δ denotes the transition function of the trie, *initial* is the root node of the trie, $L[q]$ is the maximum length of labels of paths from the root node to the target node q, p is the current node of the trie, p' is the current node of the minimal absent words trie, q' is the target node of transition function of the minimal absent words trie, $s\ell$ is the suffix link of (nonempty) state p, and *reach* is the reach status of state p. The value of reach is either equal to 0, if that state has not been reached, or equal to 1, if the state has been reached. The integer ℓ is a bounded length.

AWT($\mathcal{T}(y, \ell)$)

```
1  M ← New-Automaton()
2  List ← Empty-Queue()
3  List ← Enqueue(List, (initial[T(y, ℓ)], initial[M]))
4  while List ≠ 0 do
5        (p, p′) ← Dequeue(List)
6        for a ∈ A do
7              if (δ(p, a) = NULL) and
                 ((p = initial[T(y, ℓ)]) or
                 (δ(sℓ[p], a) ≠ NULL)) then
8                    q′ ← New-State()
9                    terminal[q′] ← TRUE
10                   Succ[p′] ← Succ[p′] ∪ {(a, q′)}
11             elseif (δ(p, a) ≠ NULL) and
                 (L[δ(p, a)] < ℓ) and
                 (reach[δ(p, a)] ≠ 1) then
12                   q′ ← New-State()
13                   Succ[p′] ← Succ[p′] ∪ {(a, q′)}
14                   List ← Enqueue(List, (δ(p, a), q′))
15 return M
```

Fig. 2 displays the minimal absent words trie of the string CAGACCGTTT that is computed by this algorithm. All terminal states represent the values of minimal absent words corresponding to the string, and the outputs are AA, AT, CT, GC, GG, TA, TC, TG, ACA, ACG, AGT, CAC, CCA, CCC, CGA, GAG, and TTTT.

Theorem 1. *The algorithm AWT computes the minimal absent words of a string of length n with the bounded length ℓ in time $O(n\times$ card $A)$, where* card *(A) is the size of alphabet set.*

Proof. The operations of the main loop, except the **for** loop in line 6, execute in constant time; this gives a time $O(n)$ for their global execution. In line 14, each operation to Enqueue does not correspond to every node in the trie; therefore it is not a function of the string size. Each operation in the **for** loop in line 6 has the total number of targets being bounded by the size of the alphabet (card A), and the cumulated time of all the executions of line 6 is O(card A). Therefore, the total time of the minimal absent words construction is $O(n\times$ card $A)$. □

3.2. Similarity/dissimilarity measure

In this subsection, we present how to compute the similarity/dissimilarity of genomic sequences using minimal absent words. We present how to find the length-weighted index of minimal absent words between two sequences. For example, let us consider sequence A and sequence B, where $A = \texttt{ATGAGTGATAGACC}$ and $B = \texttt{GTGGCTATGTTAAC}$. We define X and Y as the sets of minimal absent words of sequence A and sequence B respectively. Next, we compute the minimal absent words of A and B, and we get two sets of minimal absent words: $X = \{$AA, AGAG, AGAT, ATGAT, CA, CCC, CG, CT,

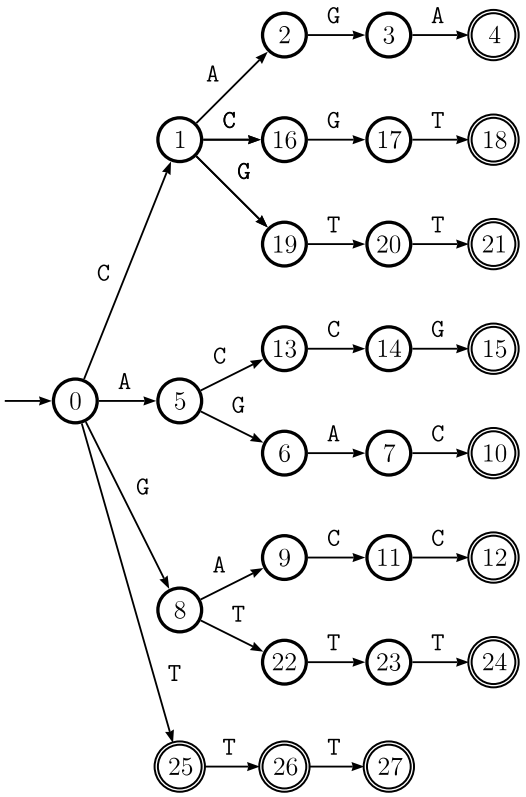

Fig. 1. Trie of CAGACCGTTT with bounded length factors is equal to 4. The initial state is 0 and terminal states are double-circled. These terminal states correspond to factors of length 4 of this string.

GAGA, GATG, GC, GG, GTA, GTGAG, TAC, TAGT, TAT, TC, TGAC, TGT, TT} and $Y = \{$AAA, AAT, ACT, AG, ATA, ATGG, ATT, CA, CC, CG, CTAA, CTG, CTT, GA, GGG, GGT, GTA, GTGT, TAC, TC, TGC, TGTG, TTAT, TTG, TTT$\}$. After that we find the symmetric difference between X and Y, i.e., $X \triangle Y = \{$AA, AAA, AAT, ACT, AG, AGAG, AGAT, ATA, ATGAT, ATGG, ATT, CC, CCC, CT, CTAA, CTG, CTT, GA, GAGA, GATG, GC, GG, GGG, GGT, GTGAG, GTGT, TAGT, TAT, TGAC, TGC, TGT, TGTG, TT, TTAT, TTG, TTT$\}$. Then we compute the length-weighted index of this example:

$$
\begin{aligned}
\sum_{w \in X \triangle Y} \frac{1}{|w|^2} &= \frac{1}{|\mathtt{aa}|^2} + \frac{1}{|\mathtt{aaa}|^2} + \frac{1}{|\mathtt{aat}|^2} + \frac{1}{|\mathtt{act}|^2} + \frac{1}{|\mathtt{agag}|^2} + \frac{1}{|\mathtt{agat}|^2} \\
&\quad + \frac{1}{|\mathtt{ag}|^2} + \frac{1}{|\mathtt{ata}|^2} + \frac{1}{|\mathtt{atgat}|^2} + \frac{1}{|\mathtt{atgg}|^2} + \frac{1}{|\mathtt{att}|^2} + \frac{1}{|\mathtt{cc}|^2} \\
&\quad + \frac{1}{|\mathtt{ct}|^2} + \frac{1}{|\mathtt{ccc}|^2} + \frac{1}{|\mathtt{ctaa}|^2} + \frac{1}{|\mathtt{ctg}|^2} + \frac{1}{|\mathtt{ctt}|^2} + \frac{1}{|\mathtt{ga}|^2} + \frac{1}{|\mathtt{gc}|^2} \\
&\quad + \frac{1}{|\mathtt{gaga}|^2} + \frac{1}{|\mathtt{gatg}|^2} + \frac{1}{|\mathtt{gg}|^2} + \frac{1}{|\mathtt{ggg}|^2} + \frac{1}{|\mathtt{ggt}|^2} + \frac{1}{|\mathtt{gtgag}|^2} \\
&\quad + \frac{1}{|\mathtt{gtgt}|^2} + \frac{1}{|\mathtt{tagt}|^2} + \frac{1}{|\mathtt{tat}|^2} + \frac{1}{|\mathtt{tgac}|^2} + \frac{1}{|\mathtt{tgc}|^2} + \frac{1}{|\mathtt{tgt}|^2} \\
&\quad + \frac{1}{|\mathtt{tgtg}|^2} + \frac{1}{|\mathtt{tt}|^2} + \frac{1}{|\mathtt{ttat}|^2} + \frac{1}{|\mathtt{ttg}|^2} + \frac{1}{|\mathtt{ttt}|^2} \\
&= \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{3^2} + \frac{1}{3^2} + \frac{1}{2^2} + \frac{1}{4^2} + \frac{1}{4^2} + \frac{1}{3^2} + \frac{1}{5^2} + \frac{1}{4^2} + \frac{1}{3^2} + \frac{1}{2^2} \\
&\quad + \frac{1}{3^2} + \frac{1}{2^2} + \frac{1}{4^2} + \frac{1}{3^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{2^2} + \frac{1}{4^2} + \frac{1}{2^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{3^2} \\
&\quad + \frac{1}{5^2} + \frac{1}{4^2} + \frac{1}{4^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{3^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{2^2} + \frac{1}{4^2} + \frac{1}{3^2} + \frac{1}{3^2}
\end{aligned}
$$

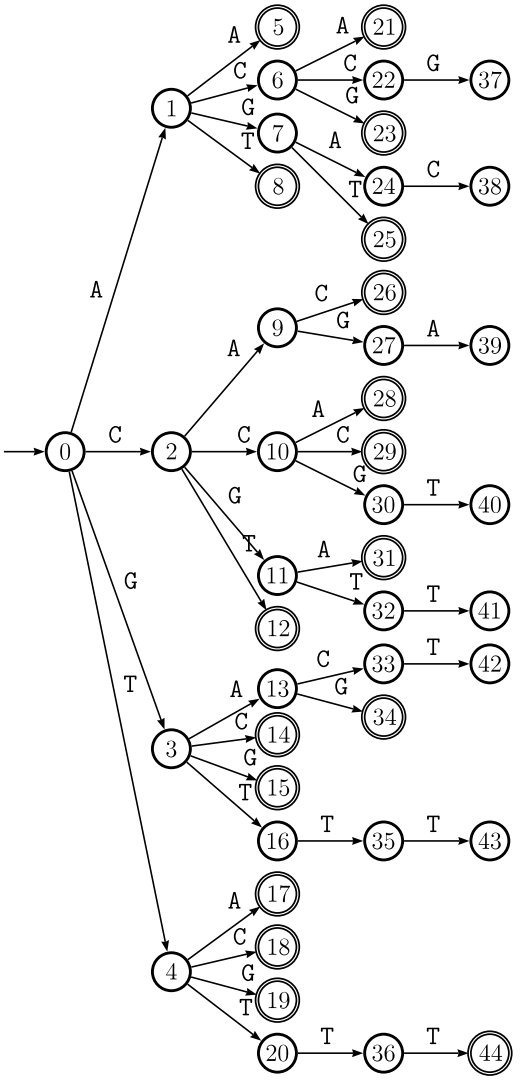

Fig. 2. Minimal absent words trie of CAGACCGTTT

$$
\begin{aligned}
&= \frac{1}{4}+\frac{1}{9}+\frac{1}{9}+\frac{1}{9}+\frac{1}{4}+\frac{1}{16}+\frac{1}{16}+\frac{1}{9}+\frac{1}{25}+\frac{1}{16}+\frac{1}{9}+\frac{1}{4}+\frac{1}{9}\\
&\quad+\frac{1}{4}+\frac{1}{16}+\frac{1}{9}+\frac{1}{9}+\frac{1}{16}+\frac{1}{4}+\frac{1}{16}+\frac{1}{4}+\frac{1}{4}+\frac{1}{9}+\frac{1}{9}+\frac{1}{25}+\frac{1}{16}\\
&\quad+\frac{1}{16}+\frac{1}{9}+\frac{1}{16}+\frac{1}{9}+\frac{1}{9}+\frac{1}{16}+\frac{1}{4}+\frac{1}{16}+\frac{1}{9}+\frac{1}{9}\\
&= 4.434167.
\end{aligned}
$$

As a result, we get the value of length-weighted index between sequence A and sequence B, 4.434167. This value represents the similarity/dissimilarity between these two sequences.

4. Results and discussion

First, in Section 4.1, we present the data set which is used for our experiments. Next, in Section 4.2, we present the experimental results for minimal absent word computation using the trie of bounded length factors. Then, in Section 4.3, we also display experimental results for the construction of phylogenies based on minimal absent words.

Table 1
Coding sequences of the first axon sequences of β-globin genes from Human, Goat, Gallus, Opossum, Lemur, Mouse, Rabbit, Rat, Bovine, Gorilla, and Chimpanzee.

Species	Coding sequences
Human	ATGGTGCACCTGACTCCTGAGGAGAAGTCTGCCGTTACTGCCCTGTGGGGCAA GGTGAACGTGGATTAAGTTGGTGGTGAGGCCCTGGGCAG
Goat	ATGCTGACTGCTGAGGAGAAGGCTGCCGTCACCGGCTTCTGGGGCAAGGTGAA AGTGGATGAAGTTGGTGCTGAGGCCCTGGGCAG
Opossum	ATGGTGCACTTGACTTCTGAGGAGAAGAACTGCATCACTACCATCTGGTCTAA GGTGCAGGTTGACCAGACTGGTGGTGAGGCCCTTGGCAG
Gallus	ATGGTGCACTGGACTGCTGAGGAGAAGCAGCTCATCACCGGCCTCTGGGGCAA GGTCAATGTGGCCGAATGTGGGGCCGAAGCCCTGGCCAG
Lemur	ATGACTTTGCTGAGTGCTGAGGAGAATGCTCATGTCACCTCTCTGTGGGGCAA GGTGGATGTAGAGAAAGTTGGTGGCGAGGCCTTGGGCAG
Mouse	ATGGTTGCACCTGACTGATGCTGAGAAGTCTGCTGTCTCTTGCCTGTGGGCAA AGGTGAACCCCGATGAAGTTGGTGGTGAGGCCCTGGGCAGG
Rabbit	ATGGTGCATCTGTCCAGTGAGGAGAAGTCTGCGGTCACTGCCCTGTGGGGCAA GGTGAATGTGGAAGAAGTTGGTGGTGAGGCCCTGGGC
Rat	ATGGTGCACCTAACTGATGCTGAGAAGGCTACTGTTAGTGGCCTGTGGGGAAA GGTGAACCCTGATAATGTTGGCGCTGAGGCCCTGGGCAG
Gorilla	ATGGTGCACCTGACTCCTGAGGAGAAGTCTGCCGTTACTGCCCTGTGGGGCAA GGTGAACGTGGATGAAGTTGGTGGTGAGGCCCTGGGCAGG
Bovine	ATGCTGACTGCTGAGGAGAAGGCTGCCGTCACCGCCTTTTGGGGCAAGGTGAA AGTGGATGAAGTTGGTGGTGAGGCCCTGGGCAG
Chimpanzee	ATGGTGCACCTGACTCCTGAGGAGAAGTCTGCCGTTACTGCCCTGTGGGGCAA GGTGAACGTGGATGAAGTTGGTGGTGAGGCCCTGGGCAGGTTGGTATCAAGG

4.1. Data set

In this subsection, we present the data sets of the first axon sequences of β-globin genes. The genes are responsible for the creation of the beta parts (roughly half) of the oxygen transport protein hemoglobin. They are from 11 species: Human, Goat, Gallus, Opossum, Lemur, Mouse, Rabbit, Rat, Bovine, Gorilla, and Chimpanzee. The coding sequences are listed in Table 1. This data set is the same set as the one used by Liu and Wang for a relative similarity measure of DNA sequences in order to build phylogenetic trees [10]. The Lempel–Ziv complexity has been applied in their approach.

The length of coding sequences of Bovine and Goat is 86, the smallest coding sequences. In contrast, the length of coding sequences of Chimpanzee is 105, the largest coding sequences. The length of coding sequences of Rabbit is 90 while the length of coding sequences of Lemur, Gallus, Human, Opossum, and Rat are 92. The length of coding sequences of Gorilla and Mouse are 93 and 94, respectively.

4.2. Experimental results of minimal absent word computation

In this subsection, we present some experimental results for minimal absent word computations, namely the growth of minimal absent words and memory size. Fig. 3 presents the growth of minimal absent words of the first axon sequences of β-globin genes from 11 genomes. The results show that the range of minimal absent words length is between 2 and 10, and Gorilla is the species which has the longest length of minimal absent words, i.e., 10. In contrast, Bovine, Gallus, Mouse, Opossum, Rabbit, and Rat are the species that have the shortest length of minimal absent words, i.e., 2. Gorilla is also the species which has the widest range of length of minimal absent words, namely between 2 and 10. The maximum length of minimal absent words for each genome is either 4 or 5. Bovine, Chimpanzee, Goat, and Human have the maximum length of minimal absent words equal to 4, while Lemur, Gallus, Gorilla, Mouse, Opossum, Rabbit, and Rat have the maximum length of minimal absent words equal to 5.

Fig. 4 displays the trend of memory size for minimal absent word computation. It shows that it is a function that grows linearly with respect to the string length.

4.3. Experimental results of phylogeny building using minimal absent words

In this subsection, we present experimental results for phylogeny construction based on minimal absent words. First, we applied the length-weighted index technique to compute the similarity/dissimilarity between organisms. Then we used a famous phylogeny building technique, the UPGMA, to build the phylogeny of the species shown in Table 1. After we have computed the length-weighted index for all pairs of the sequences in Table 1, we get the distance matrix, which is presented in Table 2.

Taking the first row in Table 2, for example, the element 11.9599 represents the value of dissimilarity between Goat and Human. Note that 8.82943 and 9.8847 in this row are closer than 12.7895, so we say that Gorilla and Chimpanzee are most similar to Human in terms of the coding sequences of the first axon of β-globin genes. Among these species, Human

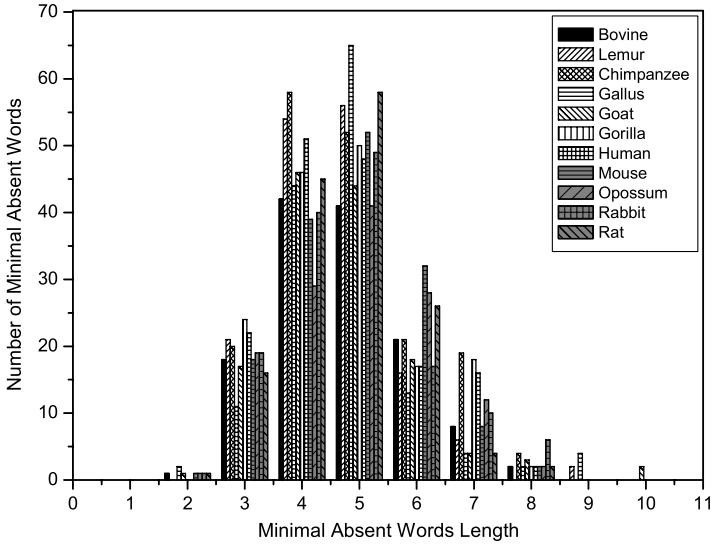

Fig. 3. Growth of minimal absent words.

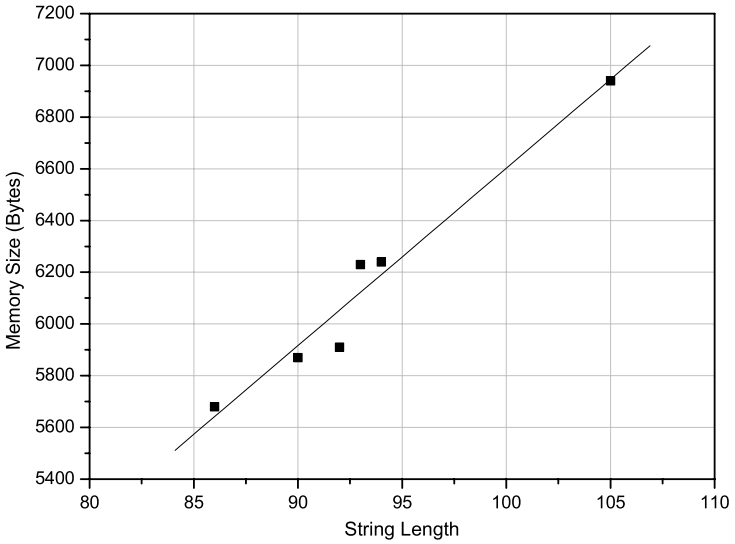

Fig. 4. Memory size required for minimal absent word computation.

and Gorilla, Human and Chimpanzee (from the first row), Goat and Bovine, Goat and Rabbit, Goat and Mouse, Goat and Gorilla (from the second row), Mouse and Bovine, Mouse and Gorilla, Mouse and Goat (from the sixth row), Rabbit and Goat, Rabbit and Bovine (from the seventh row), Rat and Goat, Rat and Bovine, Rat and Rabbit, Rat and Mouse (from the eighth row), Gorilla and Human, Gorilla and Chimpanzee (from the ninth row), Bovine and Goat, Bovine and Mouse (from the tenth row), and Chimpanzee and Gorilla, Chimpanzee and Human (from the eleventh row) are the most similar. Gallus and Opossum are always the most remote from the other species in most cases, perhaps since Gallus is the only nonmammalian representative and Opossum is the most remote species from the remaining mammals. These coincide with real biological phenomena. Besides Gallus and Opossum, Lemur is relatively more remote from the other species.

We apply UPGMA technique to construct a phylogenetic tree from the similarity/dissimilarity matrix in Table 2 and we get the phylogenetic tree which is presented in Fig. 5. Our result is similar to the result from the work of [10] that we present in Fig. 6. Both figures show that Lemur is the most remote from the other species. In both trees, Bovine and Goat are close to each others, as well as Gorilla and Human are. Alike in both trees Chimpanzee is the closest species to Gorilla and Human.

5. Conclusion

Minimal absent words in genomic sequences are an interesting and important area for studying and they provide useful information for further studies, for instance, phylogeny building. In this paper, we have defined the term minimal absent words as the set of all possible minimal absent words, and we have provided a linear-time algorithm for minimal absent word

Table 2
Similarity/dissimilarity measure of analyzed genomes.

Species	Human	Goat	Opossum	Gallus	Lemur	Mouse	Rabbit	Rat	Gorilla	Bovine	Chimpanzee
Human	0	11.96	12.79	13.33	13.70	12.11	12.09	13.35	8.83	11.90	9.88
Goat		0	12.22	11.87	12.42	11.78	11.71	12.70	11.83	8.71	12.49
Opossum			0	12.49	13.81	13.55	12.24	13.36	12.75	12.47	13.30
Gallus				0	13.55	13.08	12.07	13.96	13.62	12.51	14.04
Lemur					0	13.43	13.42	13.80	13.49	12.25	14.19
Mouse						0	12.66	12.97	11.73	11.59	12.55
Rabbit							0	12.95	12.18	11.93	12.66
Rat								0	13.39	12.81	14.22
Gorilla									0	11.78	9.40
Bovine										0	12.37
Chimpanzee											0

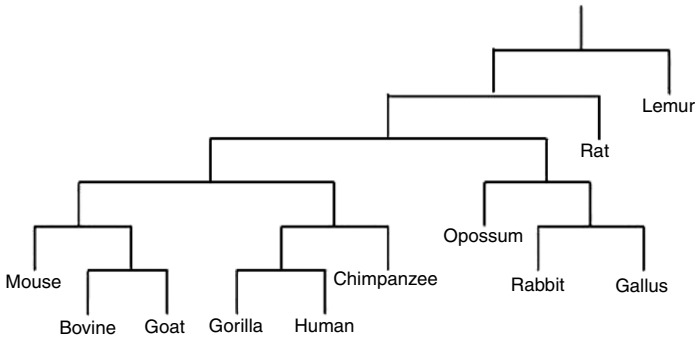

Fig. 5. Phylogenetic tree of 11 genomes based on minimal absent words.

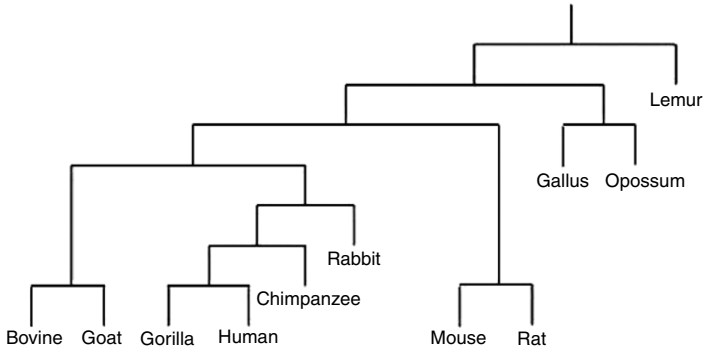

Fig. 6. Phylogenetic tree of 11 genomes from the work of [10].

computation by a trie of bounded length factors. The memory size of our approach is less than that of previous solutions. We have also applied a length-weighted index to compute the similarity/dissimilarity between genomic sequences. We have presented some properties of minimal absent words from the first axon of β-globin that contain useful information and can be applied to build a phylogenetic tree.

References

[1] C. Acquisti, G. Poste, D. Curtiss, S. Kumar, Nullomers: really a matter of natural selection? in: PLoS ONE, 2007.
[2] M.P. Béal, M. Crochemore, F. Mignosi, A. Restivo, M. Sciortino, Forbidden words of regular languages, Fundamenta Informaticae (2003) 121–135.
[3] M.P. Béal, F. Mignosi, A. Restivo, Minimal forbidden words and symbolic dynamics, in: STACS'96, 1996, pp. 555–566.
[4] H.J. Böckenhauer, D. Bongartz, Algorithmic Aspects of Bioinformatics, Springer, Berlin, 2007.
[5] M. Crochemore, C. Hancart, T. Lecroq, Algorithms on Strings, Cambridge University Press, Cambridge, UK, 2007.
[6] M. Crochemore, F. Mignosi, A. Restivo, Automata and forbidden words, Information Processing Letters (1998) 111–117.
[7] M. Crochemore, F. Mignosi, A. Restivo, S. Salemi, Data compression using antidictonaries, in: Proceedings of the I.E.E.E., 2000, pp. 1756–1768.
[8] G. Hampikian, T. Andersen, Absent sequences: nullomers and primes, in: Pacific Symposium on Biocomputing, p. 355–366, 2000.
[9] J. Herold, S. Kurtz, R. Giegerich, Efficient computation of absent words in genomic sequences, BMC Bioinformatics 9 (2008).
[10] N. Liu, T.M. Wang, A relative similarity measure for the similarity analysis of dna sequences, Chemical Physics Letters (2005) 307–311.
[11] A.J. Pinho, P.J. Ferreira, S.P. Garcia, J.M. Rodrigues, On finding minimal absent words, BMC Bioinformatics 10 (2009).
[12] M.S. Rosenberg, Sequence Alignment: Methods, Models,Concepts, and Strategies, University of California Press, California, 2009.
[13] W.K. Sung, Algorithms in Bioinformatics: A Practical Introduction, CRC Press, London, UK, 2009.
[14] Z.D. Wu, T. Jiang, W.J. Su, Efficient computation of shortest absent words in a genomic sequence, Information Processing Letters (2010) 596–601.